THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today (1) was not written for publication in a law journal and (2) is not binding precedent of the Board.

Paper No. 19

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

_____

Ex parte GARY N. HAMMOND
and
PRADEEP DUBEY

_____

Appeal No. 94-3710
Application 07/777,608[1]

_____

ON BRIEF

_____

Before HARKCOM, **Vice Chief Administrative Patent Judge**, KRASS and LEE, **Administrative Patent Judges**.

LEE, **Administrative Patent Judge**.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 1-6. No claim has

[1] Application for patent filed October 16, 1991.

been allowed.

References relied on by the Examiner

Kaplinsky	4,669,043	May 26, 1987
Cepulis	5,144,551	Sep. 1, 1992 (filed June 13, 1991)

The Rejections on Appeal

Claims 1-6 stand finally rejected under 35 U.S.C. § 103 as being unpatentable over Kaplinsky in view of Cepulis. The appellant has grouped claims 1-6 together for argument purposes in this appeal. (Br. at 4). Claim 1 is representative of claims 1-6.

An amendment to claims 1-3 was filed after the examiner's final Office action, and a further amendment to claim 1 was filed with the appellant's Reply Brief. These amendments have been entered.

The Invention

This invention is directed to maintaining coherency between a data cache and a segment descriptor cache in a

computer memory management system that uses a memory segmentation scheme. Segment descriptors, which each contain information pertaining to a particular memory segment, are stored in a segment descriptor table in main memory. When a segment descriptor needs to be loaded into the segment register, it will be loaded from the segment descriptor cache if it is present in the segment descriptor cache. If not, the segment descriptor is copied from the segment descriptor table into a data cache. The segment descriptor is also reformatted, and the reformatted segment descriptor is stored in the segment descriptor cache.

Claim 1 is directed at a method of providing coherency between the entries in the segment descriptor cache and entries in the segment descriptor table (i.e., ensuring that entries in the segment descriptor cache are not used if the corresponding entry in the segment descriptor table has changed). In claim 1, each entry in the data cache has an inclusion bit. When set, the inclusion bit indicates that the data cache entry is associated with an entry in the segment descriptor cache. When a data cache entry with a set inclusion bit is swapped out or modified, then the entire

segment descriptor cache is flushed. (Coherency is provided between the segment descriptor table and the data cache by use of a conventional method).

Claim 1, as amended, is reproduced below:

1. A method of maintaining cache coherency between a data cache and a segment descriptor cache in a memory management system of a computer, said memory management system comprising (i) a descriptor table coupled to a data processing unit for storing segment descriptors in a first format, (ii) a data cache coupled to said data processing unit for storing a plurality of data entries including segment descriptors in said first format, each of said data entries having an inclusion bit, and (iii) a segment descriptor cache coupled to said data cache and data processing unit having a plurality of segment entries for storing segment descriptors in a second format, said method comprising:

(a) providing a segment selector, said segment selector specifying a segment descriptor in a first format in a descriptor table;

(b) retrieving said segment descriptor in said first format from said descriptor table in response to said segment selector;

(c) storing said segment descriptor in said first format into a first entry in the data cache;

(d) formatting said segment descriptor in said first format into the second format;

(e) storing said segment descriptor in said second format into a first segment entry in the segment descriptor cache;

(f) setting the inclusion bit associated with said first entry in said data cache, so that said inclusion bit indicates an association between said first entry in said data cache

stored in said step (c) and said first segment entry stored in said step (e) in said segment descriptor cache;

(g) if the segment descriptor stored in the first entry in said data cache is altered by said data processing unit, checking said inclusion bit in said data cache; and

(h) flushing said entire segment descriptor cache if said segment descriptor in said first format with a set inclusion bit has been altered.

Opinion

We do not sustain the rejection of claims 1-6 under 35 U.S.C. § 103 as being unpatentable over Kaplinsky and Cepulis.

This decision is based solely on the rationale as articulated by the examiner. We do not express an opinion on rationales not articulated by the examiner.

A. Kaplinsky is not directed at maintaining the coherency of a segment descriptor cache

According to the examiner, "Kaplinsky teaches a system for maintaining cache coherency between a data cache and a descriptor cache in a memory management system of a computer comprising a descriptor table (Figure 5) and a data cache for storing segment descriptors in a first format (Figures 8A and 8B)." Examiner's Answer, at 3. In the examiner's view, the "presence bit" in Kaplinsky constitutes an "inclusion bit" for

associating a segment descriptor in the data cache with its counterpart in the segment descriptor cache. Id. at 3, 5. The examiner states that the association between two cache memories is shown "in Woodward" (sic, Kaplinsky). Id. at 6. The examiner further states that Kaplinsky's "presence bit provides an indication of the presence of required data and a corresponding association to a related cache area." Id. (emphasis in original).

Contrary to the examiner's view, Kaplinsky does not associate a segment descriptor in the data cache with its counterpart in the segment descriptor cache, as recited in step (f) of appellant's claim 1. The only reference Kaplinsky makes to a segment descriptor cache is in describing the prior art BELLMAC-32 memory management unit. See Kaplinsky, at col. 1, lines 45-48. Thus, it does not appear that the Kaplinsky system even has a segment descriptor cache.

Moreover, while Kaplinsky contains presence bits which are associated with entries in the cache, these presence bits do not indicate that the entry in the data cache is associated with an entry in another cache. Rather, as discussed below, they merely

indicate that the entry in the cache contains a copy of the data that is in the main memory.

Kaplinsky discusses presence bits in its description of semi-associative caches, which are a compromise between a tagged cache and an associative cache. Id., at col. 10, lines 13-14. The address for each entry in the cache contains L label bits and I index bits. Thus, the semi-associative cache is divided into 2^L sub-caches each containing 2^I entries. When checking to see if a particular word is in the cache, the L label bits are compared against a content addressable memory. If there is a hit, this means the sub-address space for the particular word is mapped into a sub-cache.

The sub-cache is not filled all at once; rather, each word is copied in when it is used (as per normal caching principles). Thus, there is a presence bit associated with each entry in the cache that is set only if the appropriate word has been copied from the main memory into that entry (i.e., the data in that entry is valid). The presence bits are stored in a 2^I x 1 RAM that is associated with each sub-cache. See col. 10, line 47 - col. 11, line 9.

The 2^I x 1 RAM in Kaplinsky merely contains the presence

bits for each sub-cache; it is not a separate cache, let alone the segment descriptor cache of appellant's claim 1. To the extent that the presence bit indicates an association, it is an association between the data cache entry and the entry in main memory. Thus, Kaplinsky does not disclose an inclusion bit which indicates an association between an entry in the data cache and an entry in the segment descriptor cache, as is recited in step (f) of appellant's claim 1.

B. Cepulis does not make up for the deficiencies in Kaplinsky

The examiner relies on Cepulis as teaching a segment descriptor cache management system which includes a segment descriptor cache for storing segment descriptors in a second format. Examiner’s Answer, at 4. The examiner states that Cepulis teaches checking to see if the segment descriptor has been altered and, if so, flushing the descriptor cache. Id. at 3 (citing Figure 6C of Cepulis). The examiner also relies on Cepulis as teaching a segment selector specifying a segment descriptor in the descriptor table and retrieving said descriptor in response to a segment selector. Id. at 4. The examiner concludes that it "would have been obvious to one

skilled in the art at the time the invention was made to combine Kaplinsky's inclusion (presence) bit for descriptor-data cache coherency with the segment descriptor cache selector system of Cepulis, since the resulting memory management system would provide more efficient retrieval of descriptor data . . . ." Id.

Cepulis does not make up for the deficiencies in Kaplinsky. Cepulis teaches a method for managing memory in a computer system which contains a segment descriptor cache. See col. 5, lines 5-40. Cepulis does not contain any teaching with regard to maintaining coherency between entries in the segment descriptor cache and entries in a data cache. In fact, Cepulis does not disclose or discuss a data cache. Thus, Cepulis does not teach setting a bit to indicate that there is a relationship between entries in a data cache and a segment descriptor cache as recited in appellant's claim 1.

It is the examiner's burden to establish why one having ordinary skill in the art would have been led to the claimed invention by the express teachings or suggestions found in the prior art, or by implications contained in such teachings or suggestions. See In re Sernaker, 702 F.2d 989, 995, 217 USPQ

1, 6 (Fed. Cir. 1983). Here, the examiner has not explained why the prior art would have led one of ordinary skill in the art to set an inclusion bit associated with an entry in a data cache so as to indicate an association between the data cache entry and an entry in a segment descriptor cache. The mere fact that a modification to a reference would, in hindsight, make for a more efficient system does not mean that the modification would have been obvious to a person of ordinary skill in the art. See In re Fine, 837 F.2d 1071, 1075, 5 USPQ2d 1596, 1600 (Fed. Cir. 1988).

For the foregoing reasons, we do not sustain the rejection of claims 1-6 as being unpatentable over Kaplinsky and Cepulis.

Conclusion

The rejection of claims 1-6 as being unpatentable over Kaplinsky and Cepulis is **reversed**.

REVERSED

GARY V. HARKCOM)
Vice Chief Administrative Patent Judge)
)
) BOARD OF PATENT
)
ERROL A. KRASS) APPEALS AND
Administrative Patent Judge)
) INTERFERENCES
)
)
JAMESON LEE)
Administrative Patent Judge)

Appeal No. 94-3710
Application 07/777,608

Blakely, Sokoloff, Taylor & Zafman
12400 Wilshire Boulevard
7th Floor
Los Angeles, CA 90025